

CLAIMS

What is claimed is:

1. A method of executing program code on a target microprocessor with multiple CPU cores thereon, the method comprising:
5 selecting one of the CPU cores for testing;
 performing inter-core context switching;
 executing in parallel diagnostic code on the selected CPU core and the
 program code on remaining CPU cores.
10
2. The method of claim 1, wherein the selection of the CPU core for testing utilizes an algorithm that assures testing of each of the multiple CPU cores.
- 15 3. The method of claim 2, wherein the algorithm comprises a round-robin type algorithm.
4. The method of claim 1, further comprising:
 setting a level of aggressiveness for scheduling the testing of the
20 execution units.
5. The method of claim 4, further comprising:
 applying an aggressiveness-dependent algorithm to determine when to
 schedule all available cores for execution of the program code and
25 when to schedule parallel execution of the program code and the
 diagnostic code.
6. The method of claim 1, wherein the multiple CPU cores comprise at least four CPU cores integrated onto the microprocessor integrated circuit.
- 30 7. The method of claim 1, wherein the multiple CPU cores comprise at least eight CPU cores integrated onto the microprocessor integrated circuit.

8. The method of claim 1, wherein the diagnostic code performs diagnostic operations from a test pattern comprising operations with known expected results.
- 5 9. The method of claim 8, wherein the diagnostic code compares an actual result with a known expected result.
10. The method of claim 9, wherein the diagnostic code jumps to a fault handler if the compared results are different.
- 10 11. The method of claim 10, wherein the fault handler includes code to remove a faulty CPU core from use in executing the program code.
12. The method of claim 10, wherein the fault handler includes code to
15 perform a system halt to prevent data corruption.
13. A computer-readable program product for execution on a target microprocessor having multiple CPU cores integrated thereon, the program product comprising:
20 diagnostic code configured to be executed on a selected CPU core; and program code configured to be executed on remaining CPU cores.
14. The program product of claim 13, wherein the selected execution unit
25 rotates between the multiple execution units such that each execution unit is tested.
15. A microprocessor comprising:
a plurality of CPU cores integrated on the microprocessor chip; and
inter-core communications circuitry coupled to each of the CPU cores and
30 configured to perform context switching between the CPU cores.
16. The microprocessor of claim 15, wherein each CPU core comprises a processor core and an associated local cache memory.

17. The microprocessor of claim 15, further comprising:
control circuitry coupled to the inter-core communications circuitry and
configured select a first CPU core currently in use for diagnostic
5 testing.
18. The microprocessor of claim 17, wherein the circuitry is utilized to perform
context switching between the first CPU core and a second CPU core
which is not currently in use.
10
19. The microprocessor of claim 18, wherein the microprocessor is configured
to swap external CPU numbers between the first and second CPU cores.
20. The method of claim 1, wherein the inter-core context switching includes
15 swapping virtual CPU numbers between the CPU core selected for
diagnostics and a recently-tested CPU core being put back to use.